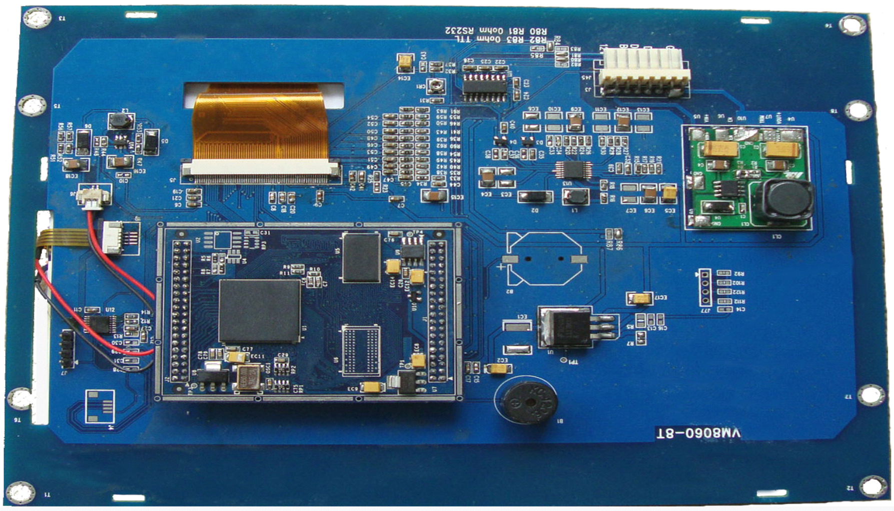
User's
Manual

GMI-6501

S 系列智能显示终端

V 1.2　　Date：2010-10-05　　产品用户手册

文档信息	
关键词	S3 图形处理模块，G 核，8 寸模组，双核心架构
概要	本文描述核心为 S3 图形处理模块的 8 寸模组的功能和使用方法



深圳市深蓝宇科技有限公司

深蓝宇科技
LANRRY TECH

版本信息		
版本号	日期	描述
V 1.2	2010-10-05	完善内容

声 明

本手册的版权归深圳市深蓝宇科技有限公司所有，并保留所有的权利。本公司保留随时更改本手册的权利，恕不另行通知。

本手册的任何一部分未经过本公司明确的书面授权，任何其他公司或个人均不允许以商业获利目的来复制、抄袭、翻译或者传播本手册。

订购产品前，请向本公司详细了解产品性能是否符合您的要求。产品并不完全具备本手册的所描述的功能，客户可根据需要增加产品的功能，具体情况请跟本公司的技术员或业务员联系。

本手册提供的资料力求准确和可靠。然而，本公司对侵权使用本手册而造成后果不承担任何法律责任。

安全使用常识 ：

- 使用前，请务必仔细阅读产品用户手册。
- 当需要对产品进行操作时请先关闭电源。
- 不要带电插拔，以免部分敏感元件被瞬间冲击电压烧毁。
- 操作者需采取防静电措施后才能触摸或进行其他可能产生静电冲击的操作。
- 避免频繁开机对产品造成不必要的损伤。

目　　录

第一章　产品简介 3
第二章　产品特点 3
第三章　产品设计流程 4
3.1 背景界面设计 4
3.2 人机交互设计 4
3.3 控制代码设计 4
3.4 机械结构设计 5
第四章　规格及尺寸 6
第五章　完整开发案例 7
5.1 项目背景 7
5.2 项目规划 7
5.3HMIDESIGNER 设计流程 9
5.4 主控 MCU 操作 15

第一章　产品简介

针对嵌入式系统开发过程中所遇到的显示界面开发难度大，开发周期长，开发成本高的特点，我公司适应市场需求，推出了图形处理能力强大，操作简单，易于控制的 S 系列图形处理单元(GPU)模块，S 系列产品具有高达 1Gb 的处理带宽以及 8MB 的 SRAM，支持 800x600 @16Bit@75Hz 及以下各种分辨率，提供各种（VESA Standard）显示标准支持，可以应用在各种需要界面显示的嵌入式系统中，帮助用户实现丰富多彩的界面风格（如 win7 风格，iPhone 风格等），满足绝大多数用户的显示需求。

第二章　产品特点

1. 设计高效——基于所见即所得的设计理念，所有背景显示部分全部使用人机界面软件 HMIDesigner 进行开发，无需编写指令与代码。
2. 串口控制——采用程序员常见的串口控制模式，串口传输协议稳定、高效，是工业应用领域最常见的接口方式之一。
3. 双核架构——主控 MCU 与 GPU 模块之间不传递背景图形图像信息，极大的降低了主控 MCU 的操作负载，主控 MCU 负责参数采集与命令控制，GPU 负责图形图像处理，双核架构，更快更稳定。
4. 精简指令——GPU 模块采用精简指令集，繁琐的图形处理工作与用户操作相隔离，绝大部分的图形交互设计仅需要 5 条以内的控制指令便可完成。
5. 标准封装——GPU 模块的控制指令全部被封装成标准 C 语言函数，函数接口参数简单，通俗易懂，使用户代码结构简单，便于维护。
6. 无缝换页——GPU 模块的各个图像页面之间实现高速切换，相对于人的感官可以做到零延时。
7. 无码触屏——触摸屏热点设置通过人机界面软件 HMIDesigner 简单实现，无需编写代码。
8. 灵活升级——采用 MCU+GPU 双核架构的嵌入式系统，用户可以随时升级 GPU 端的界面显示风格，升级界面显示尺寸，而 MCU 端的设计代码却完全不用有任何变动。

第三章　产品设计流程

3.1　背景界面设计

通常，嵌入式系统图形界面均由设计人员使用图形用户接口函数（GUI 函数）来实现，设计难度高，耗费时间长，一个普通的图标就需要一天的时间来设计、修改、绘图，画面效果也常常难以尽如人意。现在，使用我公司 S 系列 GPU 模块，用户可以在显示终端轻松实现各种丰富细腻的图片特效与立体效果，更容易实现各种流行界面风格（如 WIN7，iPhone，Mac 等）。

为达到最好的图像表现，建议用户使用专业的图像处理软件（如 Photoshop）设计图片，对图片效果要求严格的用户还可以聘请专业的美工来设计图片。

界面设计时请注意：

1)　图片颜色深度设定为 16 位色或以下。

2)　图片分辨率大小根据客户所选液晶屏分辨率大小设定。

3.2　人机交互设计

人机交互设计是界面设计的一部分，主要完成用户设备参数的界面表现部分，主要工具为我公司开发提供的专业设计软件 HMIDesigner。本软件采用组态控件式设计，将用户常用的设计方案分类归纳合并，提取出客户完整的功能需求并加以实现，提供波形控件、进度条控件、按钮控件等多种功能完善的组态控件。高度集成化的控件大大降低了用户的工作量，提高用户的工作效率。

首先，使用 HMIDesigner 软件新建方案、工程，设置工程属性参数，导入预先设计好的背景界面；

其次，在背景界面上添加客户需要的控件，并按系统需求设置控件属性参数；

最后，用串口线将 GUI 模块与电脑主机相连，依次执行联机、编译、下载，将设计好的工程下载到 GUI 模块。

设计过程中，强烈建议用户随时保存设计好的方案工程，以备将来修改或升级。

3.3　控制代码设计

用户控制代码通常实现设备数据采集，数据运算，命令控制等功能，相对主控 MCU 来讲,GPU 模块作为单独的图形图像处理终端，在整个嵌入式系统中，可以被看做一个单独的外部设备来处理，所以主控 MCU 需要在控制代码中加入对 GPU 模块的控制命令。

GPU 模块所能识别的全部指令，被称作 GPU 的指令集，通过采用高度集成化的封装模式，GPU 的指令集规模非常的小。

为方便用户使用，GPU 模块的指令又被二次封装成标准 C 语言函数，对每个组态控件的操作都有对应的函数来实现。如波形控件对应的函数为：void VH_WaveUpdate(U8 WidgetId,U8 DataLen,U8 *pData)；进度条控件对应的函数为：void VH_EpbarUpdate(U8 WidgetId,U8 Data)。

用户只需要在程序的合适位置调用这些函数，便可以实现对组态控件的实时控制与状态更新。

代码调试完毕，整个界面开发工作已基本完成，强烈建议用户将设计好的程序源码归档保存，以备将来修改或升级。

3.4 机械结构设计

为方便用户使用，我公司提供背板式的 GPU 模组，即包含有液晶屏，安装底板，核心 GPU 模块的三位一体模组，用户可以根据 GPU 模组尺寸设计安装外壳。如用户已有正在使用的外壳，我公司也可为用户特别订制专属尺寸的安装模组，具体操作流程请与业务人员洽谈。

第四章　规格及尺寸

型号		GMI-6501
产品特性	液晶屏	8″TFT 真彩液晶显示
	分辨率	800×600
	显示颜色	65535 色（16bit 真彩）
	亮度/背光灯	250/LED(软件可调亮度）
	背光寿命	50000 小时
	触摸屏	电阻式（自带校准程序）
	图片格式	JPEG
	电压	12V，电源可选配范围 6V～18V
	功耗	12V/350mA
	核心模块	S3
	核心频率	125MHz
	显存容量	8MB
	显存带宽	1Gb
	存储器容量	8MB
产品规格	结构	底板固定
	颜色	---
	显示区尺寸	162×121.5(mm)
	开孔尺寸	
	净重	
接口	串口	1×RS232（速率最大达 115200bps，可调）
	键盘	选配
环境条件	工作温度	-20℃-70℃
	工作湿度	10～90%RH
	储存温度	-30℃-80℃
	储存湿度	8～90%RH
软件	组态控件	HMIDesigner 专用开发软件

第五章　完整开发案例

5.1　项目背景

本项目是用在油井憋压测量中的一个测量设备，在采集、测控项目中一般都需要实时显示波形图及当前的各种平均量。本项目即要求在界面上显示憋压的压力值，方便现场工程师查看。

设备采集一个压力传感器的压力值，压力传感量 0-10MPa，压力传感器 AD 量化值输出 0-4096；采集时间 3.5 分钟以内。

工作流程有：

现场工程师点击触摸屏上的“开始采集”按钮启动主控 MCU 进行采集工作。

主控 MCU 在采集过程中实时将波形和时间等采集数据发送给 GPU 显示。

现场工程师点击触摸屏上的“停止采集”按钮停止主控 MCU 的采集工作。

主控 MCU 在采集达到指定时间后自动停止采集工作，并通知 GPU 模块。

5.2　项目规划

基于项目需求，本案例选用波形、数值框、按钮等控件完成憋压测量人机界面，主控 MCU 向 GPU 模块发送波形控件和数值控件中需要显示的值，通过 GPU 模块的触摸屏控制主控 MCU 的各种动作。本项目在 7 寸模组 GMI-6401 下实现，显示分频率为 800x480 像素。

波形

GMI-6401 中屏幕高度为 480 个像素，0-4096 的数据全部显示，超出了范围，同时波形控件中支持图像放大，不支持图像缩小，因此将 0-4096 缩小 10 倍为 0-409.6，取整为 0-410；在纵向使用 410 个像素。最大采集时间为 3.5 分钟，共 210 秒，在横向 800 个像素中可以使用 630 个像素显示时间轴。

即波形窗体为 410x630 像素。

数值

本项目中可以从波形显示中确认采集时间，但是为了更为直观的显示采集的进行时间，设计一个数值用来显示采集时间。

按钮

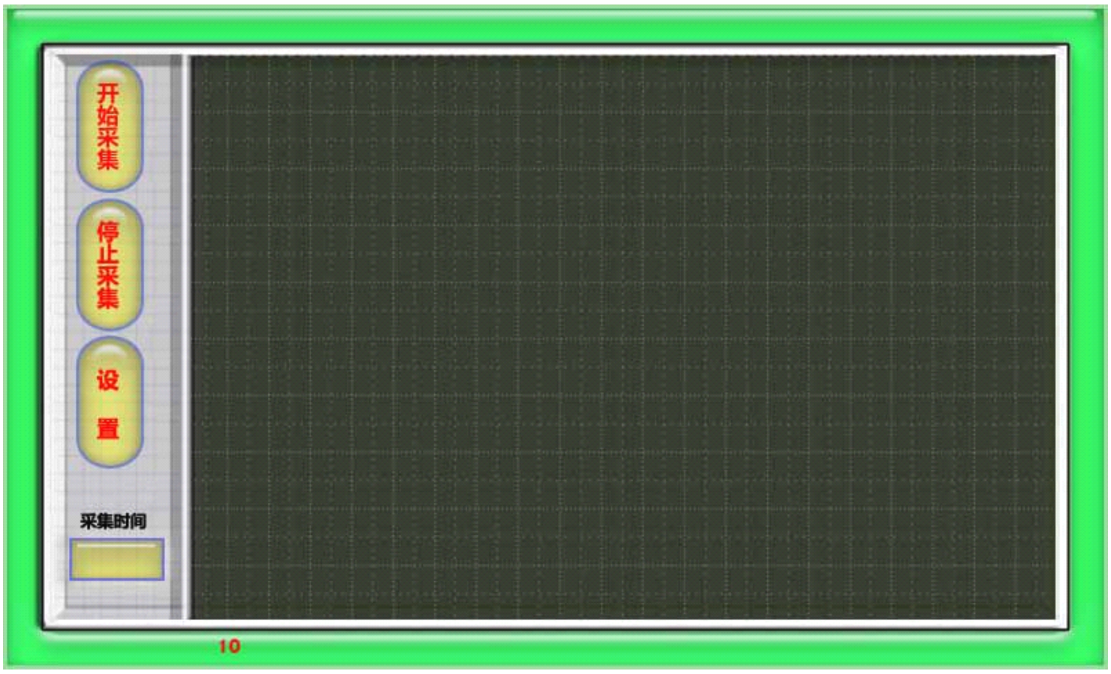
本项目中通过触摸屏作为人机界面的输入，主控 MCU 收到触摸屏的指令后采集数据并送给 GMI-6401 模组显示。因此本项目中添加《开始采集》《结束采集》《设置》三个按钮。为了简单示意，暂时只实现《开始采集》/《结束采集》两个按钮。同样《开始采集》《结束采集》两个指令可以在同一个按钮上实现，修改主控 MCU 的代码即可。

图片风格

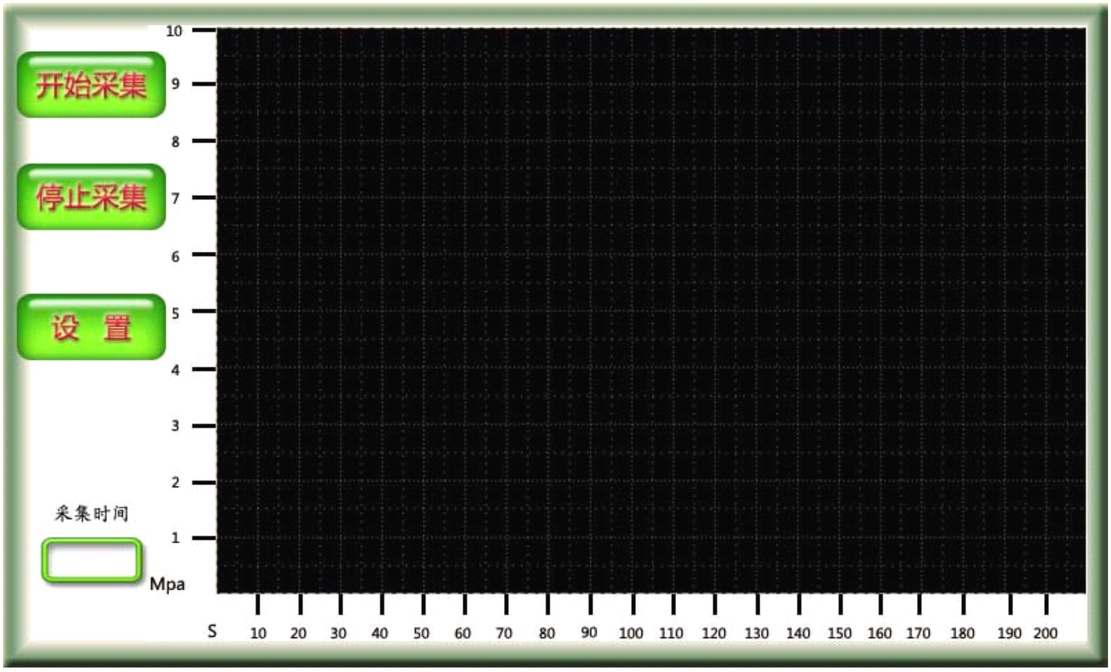
基于以上的项目规划设计画面的整体风格及背景图片。

下图设计了两种风格的界面图片，任选一张使用即可，本示例采用图片 2 作为主画面。。GPU 模块显示的每一副图像，习惯上称为“画面”或“页面”，有时也简单地称作“图片”。

图片 1



图片 2



5.3 HMIDesigner设计流程

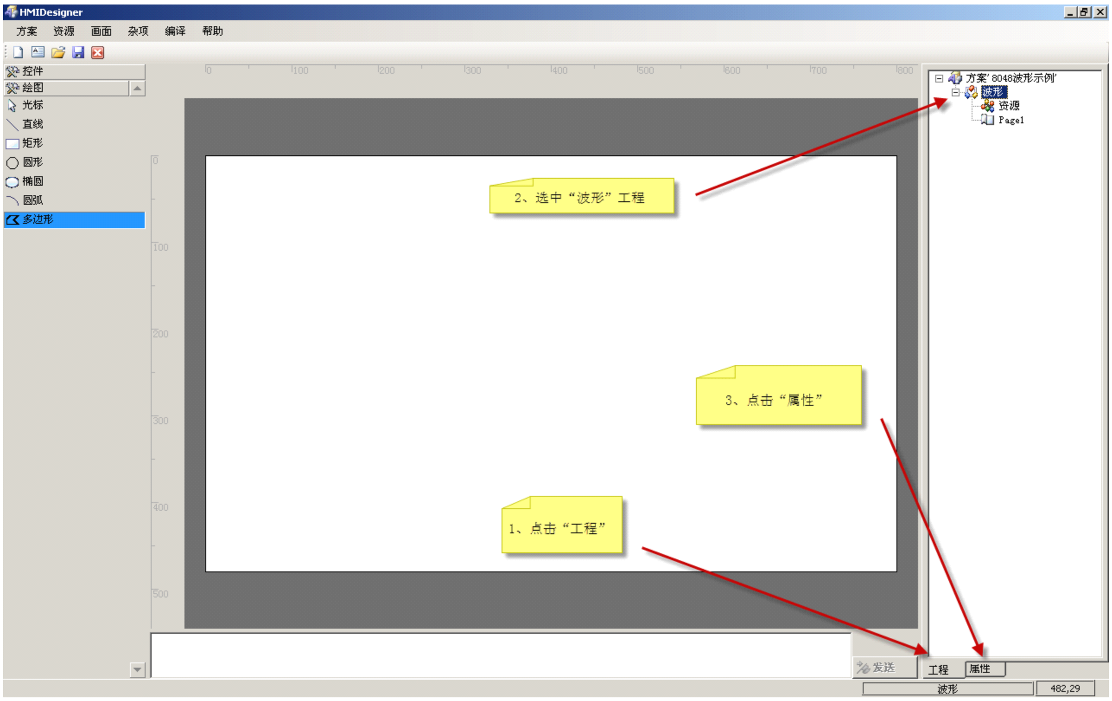
以下将详细介绍使用 HMIDesigner 软件设计界面的流程。

1. 启动 HMIDesigner，并新建方案。

    a) 定义方案名称，以及工程名称，分别为“8048 波形示例”，“波形”；新建方案的同时新建了一个工程，在同一个方案下可以定义多个工程文件。

    b) 选择板卡类型，选择通信的串口，选择保存路径。

2. 设置工程属性。



a) 点击“工程”选项卡

b) 选中“波形”工程

c) 点击“属性”选项卡，在其中设置当前工程的属性：

i. 波特率：此处的串口波特率为工程实现完成后与主控 MCU 之间通信的串口波特率，支持从 2400 到 115200 各种波特率。本项目与主控 MCU 通信采用 115200，因此选择 115200 速率。

ii. 支持触摸屏：本工程支持触摸屏，因此选择“True”

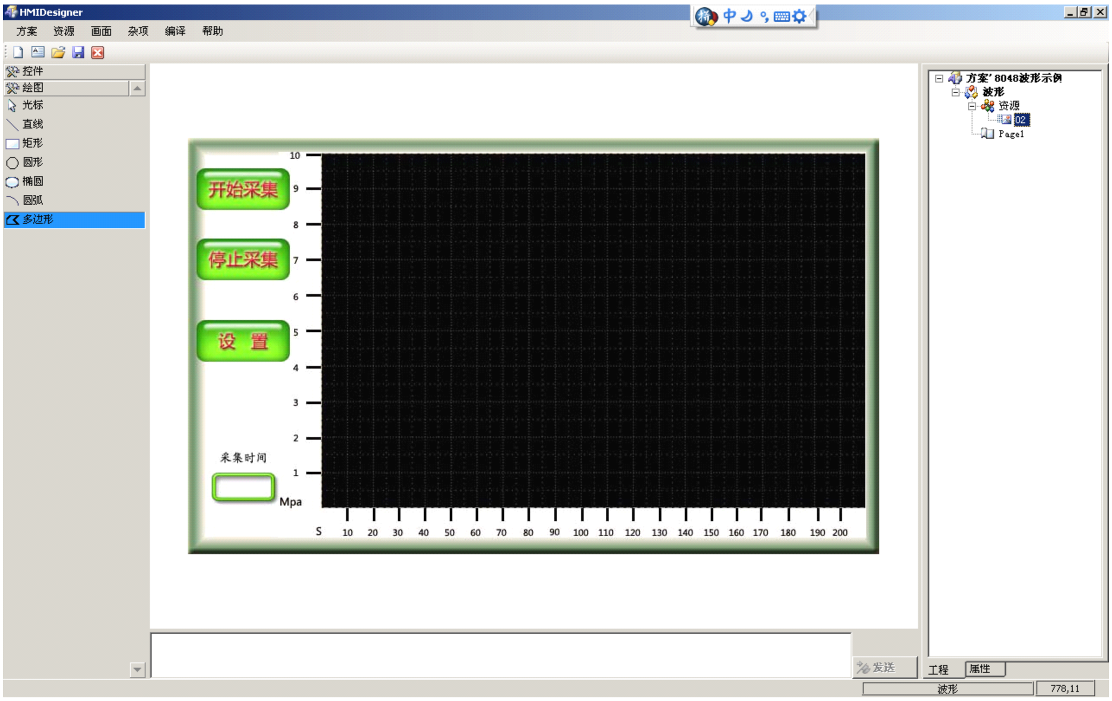
iii. 触摸屏模式：触摸屏模式选择“C”，即触摸屏可以将信息通过控件发送给主控 MCU。

到此工程属性设置完毕。

3. 添加图片

a) 选择《资源》/《导入图片》命令，弹出对话框，选择设计好的图片 2；

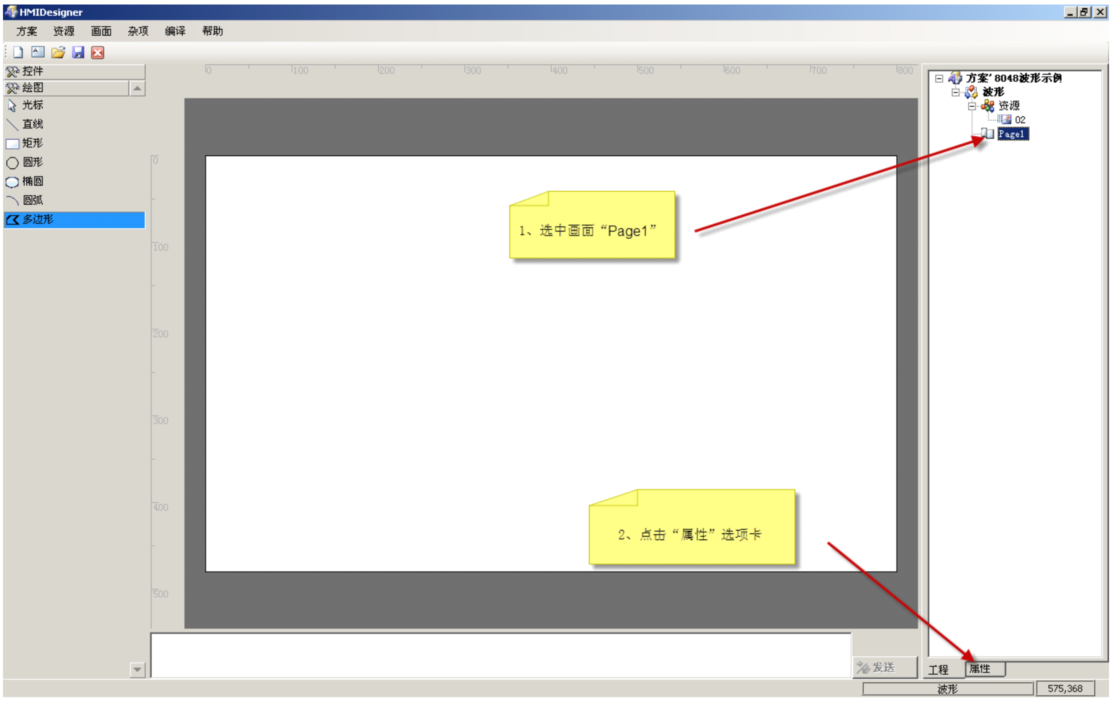
b) 使用的图片已经在资源栏目中，选择可以预览。



4. 创建画面：

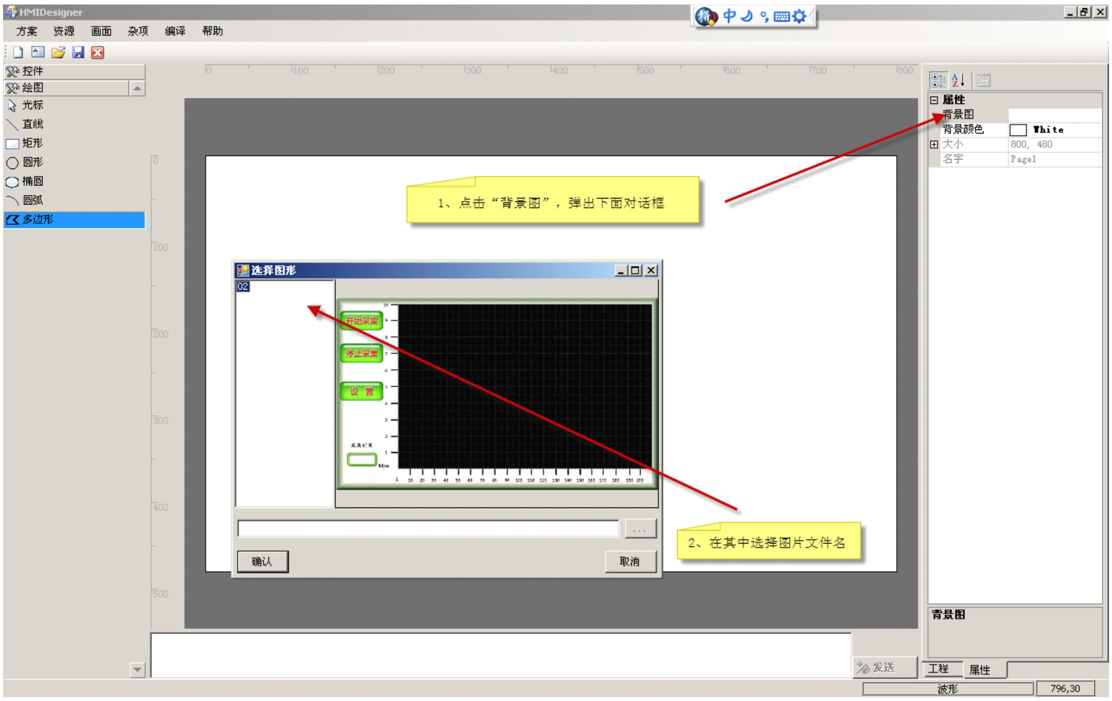
工程默认打开时会创建一个画面“Page1”，本设备仅需要一个画面进行显示，因此无需再创建画面。

5. 设置画面



在《工程》选项卡中选中画面 1“Page1”，然后点击《属性》选项卡，设置“Page1”的属性。

点击背景图，选择导入的图片文件，《确认》后，图片“02”将作为背景图在“Page1”中显示。



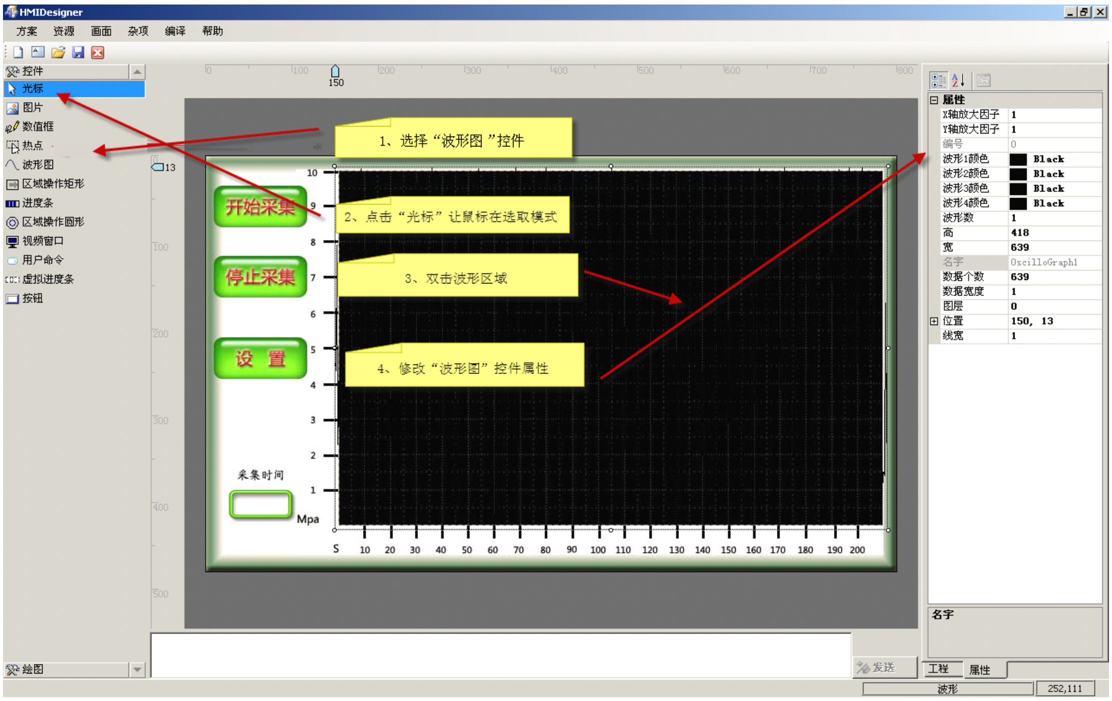
6. 添加并设置波形控件

a) 选择“波形图”控件

b)　点击“光标”让鼠标在选取模式

c)　双击波形区域

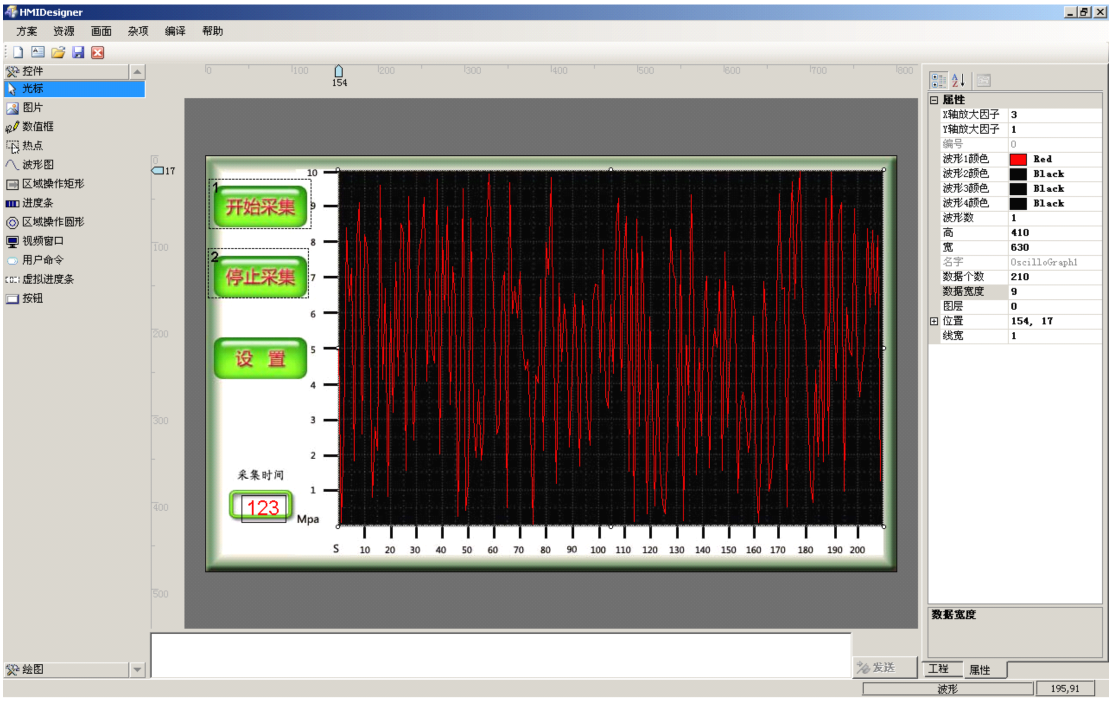
d)　修改“波形图”属性。



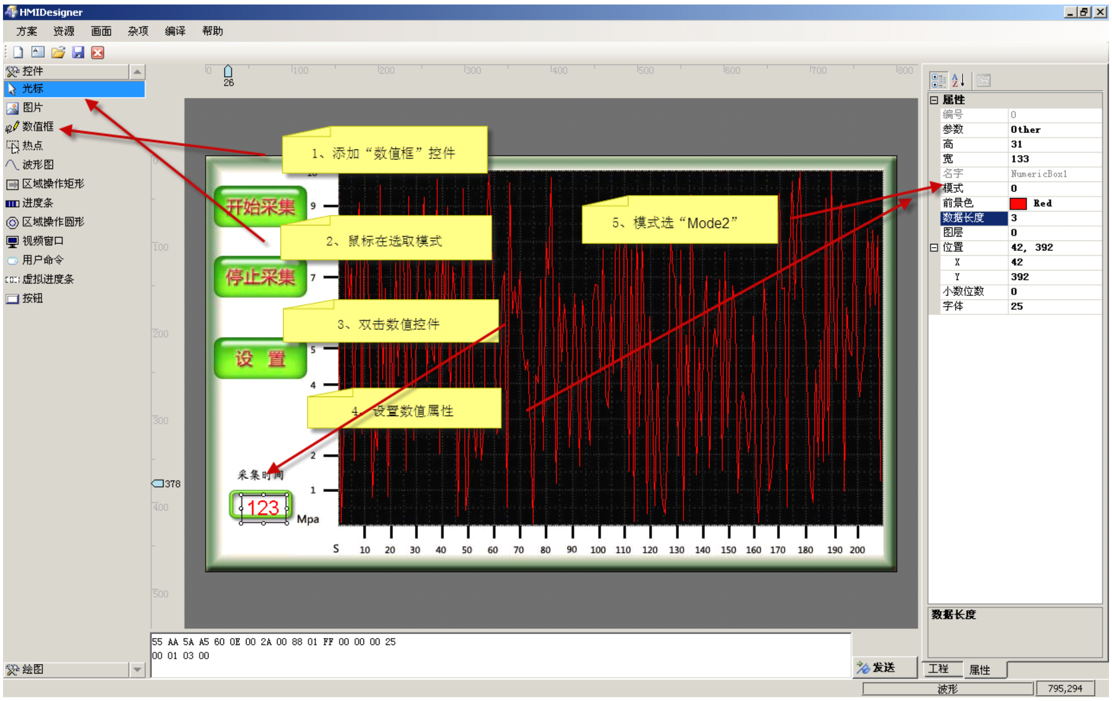
“波形”控件的主要设置如下属性：

- X 轴放大因子
- 波形颜色
- 波形数量（一个波形控件最多可以显示 4 条波形）
- 数据个数（即当前设计中波形控件中需要显示的样点数，本设备为 210 个样点）
- 位置等。

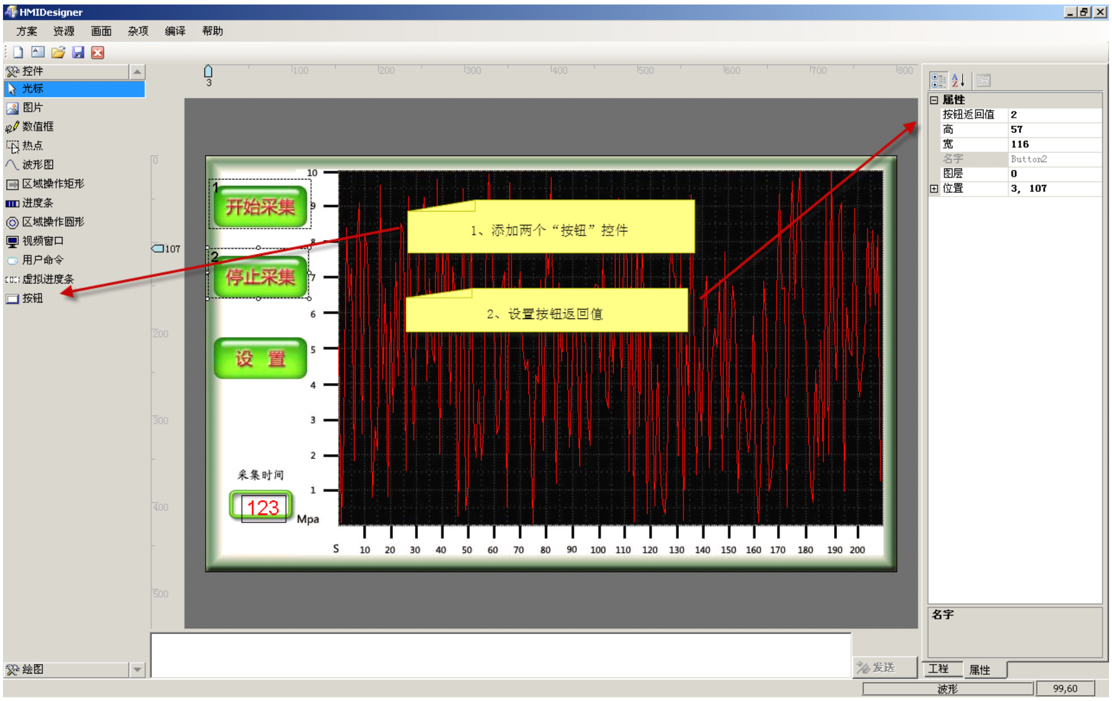
可以按照下图参数；



7. 添加并设置数值控件



8. 添加并设置按钮控件



9. 设置完毕之后，点击《编译》/《编译》命令，对所有设置进行编译，如果有设置错误的地方软件将自动提醒，编译成功后保存当前工程。

10. 联机：点击《编译》/《联机》命令，进行联机测试。联机成功则弹出对话框“联机测试正常，可以进行后续操作”。

11. 下载：点击《编译》/《下载》命令，将设计下载当 GPU 模块中。

12. 重启模组 GMI-6401 后，即运行当前的设计。

5.4 主控MCU操作

在本项目中使用主控 MCU 进行控制，模组 GMI-6401 本身作为从设备，因此在章节中将对主控 MCU 的程序设计进行说明。

使用 GPU 模块作为人机界面时，主控 MCU 对于人机界面的操作只有当前的各种交互数据，不包括各种背景图形图像信息，背景图形图像的信息由 HMIDesigner 已经全部设计完成。

主控 MCU 的类型

主控 MCU 可以是具有串口通信能力的任何 MCU.

主控 MCU 流程与 GPU 的交互命令规划

通过设备的工作过程规划主控 MCU 与 GPU 的交互过程。

- GPU 启动完成后，主动向主控 MCU 发送指令，声明 GPU 进入工作状态(GPU 准备好)。此时第一条声明有可能主控 MCU 并未接收到，因为无法确认主控 MCU 与 GPU 各自的启动时间。

- 主控 MCU 启动完成后，向 GPU 发送测试指令，GPU 收到测试指令后回应，则双方建立通道成功。
- 主控 MCU 进入等待状态，等待触摸屏按下“开始采集”命令。
- 主控 MCU 收到采集指令后，采集数据并向 GPU 发送数据。
- 主控 MCU 收到触摸屏按下“停止采集”命令后，即停止数据采集。

分析以上流程，GPU 的动作有：

1、主动发送启动正常指令，向主控 MCU 声明。

2、响应主控 MCU 的状态查询。

3、发送按键键值。

4、更新各个控件的数值。

主控 MCU 的指令有：

1、VC_SayHello:询问 GPU，如果准备后则建立通道；

2、VH_WaveUpdate:更新波形；

3、VH_ NuboxUpdate:更新“采集时间”数值框；

4、响应 GPU 发送的按键键值，并产生相应动作。

主控 MCU 操作库

GPU 中将所有人机界面的操作都封装成函数库，用户对于各种显示的操作仅仅通过调用相应的库函数即可完成。

HMIDesigner 设计软件在当前的工程下自动生成一个名为“vmcu_dfn.h”的.h 文件。同时我们提供“vcmd.c”和“vcmd.h”两个文件。

其中对于 GPU 的控制部分（即主控 MCU 的发送部分）完全为标准函数，用户直接调用即可，用户仅需要修改 vcmd.c 文件中的 VC_UartTxByte 函数而符合自己当前选定的 MCU 即可。

用户需要修改主控 MCU 的接收（即 GPU 发送过来的数据）函数，以满足当前系统的设计。

PC 下实现示例

本项目通过 PC 机模拟用户的主控 MCU 来实现设计。

- 修改 VC_UartTxByte 函数，支持标准 PC;

```
void VC_UartTxByte(U8 TxByte){
	while(!(0x20 & inportb(COM_PORT+5)));
```

```
            outportb(COM_PORT+0,Byte);
    }
```

- 修改串口终端函数，支持标准 PC。

```
    void interrupt int_com(...)
    {
     static unsigned char datanum=0;
     while((inportb(COM_PORT+5) & 0x01)!=0)
     {
        donot=1;//MaskIRQ7();
        datanum=inportb(COM_PORT);
     if ((HeadCnt==0) && (datanum == 0xa5)){
          HeadCnt=1;
          }
     else if ((HeadCnt==1)&&(datanum == 0xa5)){
       bufptr = 0;
          HeadCnt =2;
          }
     else if ((HeadCnt==2))
       HeadCnt = 3;
     else if ((HeadCnt==3))
        HaveData = 1;

      if ( HeadCnt >= 2 ){
        commbuf[bufptr]=datanum;
          if ( bufptr > 32 ) bufptr = 0;
          else
            bufptr++;
        }
        donot=0;
      }
     outportb(0x20,0x20);
    }
```

- 使用随机数模拟主控 MCU 采集好的数据，做主控 MCU 主程序

```
void main(int argv, char *argc[]){
  定义变量
  初始化
     主循环
          while(1){
          if(HaveData){//如果有串口中断
```

```
    printf(" %x %x %x \n",commbuf[0],commbuf[1],commbuf[2]);

    if ( commbuf[2] == 0x71 )//VSD 发送来的对于波形更新函数的响应
      Rspond = 1;
    else if ( commbuf[1] == 0xA3) //VSD 发送过来的按键键值标识
      ProtocolID = commbuf[2];
      printf("ID %x Rspd %x",ProtocolID,Rspond);

    HaveData = 0;
    HeadCnt = 0;

  switch( ProtocolID){
         case 0x01:
            for(ii=0;ii<4;ii++)
            {
              SendData.iData = rand()%410;//准备数据（模拟的采集数据）
              Wave[ii*2] = SendData.half.hh;
              Wave[ii*2+1] = SendData.half.ll;
            }

            if( Rspond)
            {
            tcnt++;
            printf("tcnt %x \n",tcnt);
            delay(200);
            VH_NuboxUpdate(VH_NUBOX_NumericBox1,tcnt);//更新采集时间
            delay(200);
            VH_WaveUpdate(VH_WAVEG_OscilloGraph1,8,Wave);//更新波形，每次更新 8 个样点，本函数每次
可以更新 1-410 个样点，任意选择。
            printf("SendOk.. ");
            Rspond = 0;
            }
            break;
        case 0x02:
          Rspond = 1;
          break;
        default:
        break;
      }
    }
```

```
            }
    }
```

- 通过以上的修改即完成了本项目主控 MCU 的工作。

使用 GPU 显示平台开发人机界面时，整个过程中开发工程师无需了解各种协议、指令等，主控 MCU 对于 GPU 的操作仅通过修改接口函数即可，研发工程师始终关心当前设备当中的采集样点，而显示部分全部交由 GPU 完成，极大的方便了产品的更新换代，缩短了上市周期。

需要完整的测试代码，请联系售后技术支持人员。